NYO-1480-173

Courant Institute of
Mathematical Sciences

AEC Computing and Applied Mathematics Center

Numerical Solution of the Boltzmann Equation

Alexandre Joël Chorin

AEC Research and Development Report

Mathematics
April 1971

New York University

UNCLASSIFIED

AEC Computing and Applied Mathematics Center
Courant Institute of Mathematical Sciences
New York University

Mathematics NYO-1480-173

NUMERICAL SOLUTION OF THE BOLTZMANN EQUATION

Alexandre Joël Chorin

Contract No. AT(30-1)-1480

UNCLASSIFIED

Abstract

A numerical method for solving the full non-linear Boltzmann equation is presented, and applied to the problem of shock structure in a gas of elastic spheres. The success of the method hinges on the systematic use of Gaussian quadrature and Hermite interpolation.

Introduction. The Boltzmann equation describes the evolution of the one-particle distribution function $f = f(\underline{x},\underline{u},t)$, where $\underline{x}$, with components (x_1, x_2, x_3), is the position vector, $\underline{u}$ with components (u_1, u_2, u_3), is the velocity vector, and t is the time. In the case of a gas of elastic spheres it has the form

$$(1) \qquad \frac{\partial f}{\partial t} + (\underline{u}\cdot\underline{\nabla}_x)f + \frac{1}{m}(\underline{F}\cdot\underline{\nabla}_n)f = \frac{\sigma^2}{2} \int |\underline{V}.\underline{e}|(f_+f'_+ - ff')\underline{du}'d\omega$$

where m is the mass of a particle, σ its radius, $\underline{\nabla}_x$ denotes the gradient operator with respect to the $\underline{x}$ variables, $\underline{\nabla}_u$ denotes the gradient operator with respect to the $\underline{u}$ variables, $\underline{F}$ is the external force, $\underline{e}$ is a unit vector pointing in the direction of the solid angle element $d\omega$, $\underline{V} = \underline{u}' - \underline{u}$, a bar under a symbol denotes a vector quantity, and

$$f = f(\underline{x},\underline{u},t)$$
$$f' = f(\underline{x},\underline{u}',t)$$
$$f_+ = f(\underline{x},\underline{u}^+,t)$$
$$f'_+ = f(\underline{x},\underline{u}^{+'},t)$$

where

$$\underline{u}^+ = \underline{u} + (\underline{V}\cdot\underline{e})\underline{e}$$
$$\underline{u}^{+'} = \underline{u} - (\underline{V}\cdot\underline{e})\underline{e} \; .$$

$\underline{u}^+$, $\underline{u}^{+'}$ are the velocities before collision of those spheres which after collision have the velocities $\underline{u}$ and $\underline{u}'$. Analogous expressions can be written for other kinds of interparticle force. For an elementary discussion of this equation, see [14]; for a thorough

discussion see e.g. [2] and [5]. The right hand side of equation (1) will be called the collision integral.

It is the purpose of this paper to present a numerical algorithm for solving Equation (1) and to apply it to the study of the structure of a shock in one space dimension. Generalizations of this method to problems involving very strong shocks, more space dimensions, and other molecular models, will be also discussed. It will be seen that the solution of the shock problem provides a key to the solution of the other problems; the main difficulty has been overcome in the program discussed in this paper. Furthermore, the numerical solution provides insight into some approximate procedures, in particular Grad's thirteen moment approximation [6] and Mott-Smith's bimodal approximation [12].

Unlike the work presented here, most previous numerical treatments of the Boltzmann equation relied on a Monte-Carlo technique; some of these treatments are ingenious and interesting, but none can be considered accurate. See [1], [8], [9], and [13]. Reference [9] is particularly helpful.

For any function $\phi(\underline{x},\underline{u})$, let $\overline{\phi}(\underline{x})$ denote the integral

$$\overline{\phi}(\underline{x}) = \int \phi(\underline{x},\underline{u}) f(\underline{x},\underline{u}) \underline{du} \ .$$

Some of the quantities of interest in the solution of the Boltzmann equation are the following moments of f: the density $\rho(x) = \overline{1}$, the mean velocity $\overline{u}$, the pressure $p = \frac{1}{3}\rho\overline{w^2}$, where $\underline{w} = \underline{u} - \underline{\overline{u}}$, the temperature $T = p/\rho R$, where R is the universal gas constant, the

pressure tensor $p_{ij} = \rho\overline{w_i w_j}$, and the heat flux vector $\underline{S} = \frac{1}{2}\rho\overline{\underline{w}^2\underline{w}}$.
Other quantities of interest are the Boltzmann H function

$$H = \int f \log f \, d\underline{u} \quad ,$$

and in the shock wave problem, various geometric parameters which characterize the shock.

In the case of a gas of elastic spheres, the mean free path is

$$\ell = 1/(\sqrt{2}\, \pi \, \rho \, \sigma^2) \ .$$

We shall now specialize equation (1) to a form appropriate to the shock problem. Let T_0 be a reference temperature, and u_0 a reference thermal velocity, $u_0 = \sqrt{2RT_0}$. Let τ be a collision time, $\tau = \ell/u_0$, and ρ_0 a reference density. Introduce the non-dimensional variables

$$\underline{x}^* = x/\ell \ , \ \underline{u}^* = \underline{u}/u_0 , \ t^* = t/\tau \ , \ f^* = u_0^3 f/\rho_0 \ ;$$

substitute them into equation (1) and drop the stars. Furthermore, pick units in which a reference mean free path $1/(\sqrt{2}\,\pi \ \rho_0\,\sigma^2)$ is 1. Assume $\underline{F} = 0$, (no external forces), and allow f to depend only on one space variable $x_1 = x$, and two velocity variables $u_1 = u$ and u_r, where u is in the direction of x and u_r is in a direction orthogonal to u. These assumptions imply that the flow is invariant

under rotation around the x-axis. Under these assumptions $f = f(x,u,u_r,t)$ satisfies

$$(2) \quad \frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = \int_0^\pi d\phi \sin\phi \int_0^\pi dX \int_{-\infty}^{+\infty} du' \int_{-\infty}^{+\infty} du_r' (f_+ f_+' - ff')\cdot|\underline{V}\cdot\underline{e}|/2\sqrt{2}\,\pi$$

where

$$\underline{e} = (\cos\phi,\ \sin\phi\cos X,\ \sin\phi\sin X),$$

$$\underline{u}' = (u', u_r'\cos X,\ u_r'\sin X),$$

$$\underline{u} = (u,\ u_r,\ 0),$$

$$\underline{V} = \underline{u}' - \underline{u},$$

$$\underline{u}^+ = \underline{u} + (\underline{V}\cdot\underline{e})\underline{e} = (u^+, u_2^+, u_3^+),$$

$$\underline{u}^{+'} = \underline{u} - (\underline{V}\cdot\underline{e})\underline{e} = (u^{+'},\ u_2^{+'},\ u_3^{+'}),$$

$$u_r^+ = \sqrt{u_2^{+2} + u_3^{+2}}\ ,$$

$$u_r^{+'} = \sqrt{(u_2^{+'})^2 + (u_3^{+'})^2}\ ,$$

$$f' = f(x,u',u_r'),$$

$$f = f(x,u,u_r),$$

$$f_+ = f(x,u^+,u_r^+),$$

$$f_+' = f(x,u^{+'},u_r^{+'})\ .$$

This is the form of the equation we shall use below, although the method of solution applies to the general equation (1) as well.

Principle of the method. There are several major difficulties in the solution of (1) or (2). The function f depends on a relatively large number of independent variables-three plus time in the case of one-dimensional flow, six plus time in the general case-so that if (2) is replaced by a system of algebraic equations, their number will be large. The presence of the fourfold non-linear integral ensures that the algebraic equations will be not only numerous, but also very cumbersome. Efficient computation is clearly needed. Another difficulty stems from the nature of the collision term, more specifically, from the integration over the angular variables. Suppose f is represented by a discrete set of values assumed on a discrete set Z of points in phase space. The integration over u', u_r' becomes a sum over the values assumed by f on Z. The integration with respect to θ, X becomes a sum over a discrete set Θ of values of θ, X. For any reasonable choice of Z and Θ, the arguments of f_+, f_+' will include points not in Z. Thus interpolation, both accurate and stable, between values of f on Z, will be required. (Monte-Carlo methods avoid the interpolation problem at a heavy price in accuracy.)

These difficulties can be resolved as follows: once one resigns oneself to the need for interpolation, there is no need to identify Z with the nodes of a regular mesh. One can then evaluate f at points $(x_k, u_i, u_{r,j})$ where the u_i, $u_{r,j}$ are at one's disposal. In particular, one can choose u_i, $u_{r,j}$ to be the roots of a polynomial $P_N(u)$, where P_N is the N-th degree member

a sequence of polynomials P_0, $P_1,\ldots,P_n,\ldots$, orthogonal with respect to a weight $W(u)$. With this choice of u_i, $u_{r,j}$ one can interpolate between the values of f at these points using the orthogonal polynomials $P_n(x)$. Such interpolation is stable; see details below. Furthermore, with this choice of integration points, integrals can be evaluated by an appropriate variant of Gaussian quadrature. In the case of the Boltzmann equation, it is natural to use the Hermite polynomials $H_n(u)$ given by

$$H_n(u) = (-1)^n c_n \, e^{u^2} \frac{d^n}{du^n} e^{-u^2} \; , \quad c_n = (2^n n!)^{-1/2} \; ,$$

which are orthonormal with respect to the weight $W(u) = \pi^{-1/2} e^{-u^2}$, i.e.

$$\pi^{-1/2} \int H_n(u) H_m(u) e^{-u^2} du = \delta_{n,m} \; ;$$

$\delta_{n,m}$ the Kronecker delta. The set $\{H_n(u)e^{-u^2/2}\}$ is complete in $L_2(-\infty,+\infty)$. See [10].

The preceding remarks lead to the following step-by-step procedure for solving (2): Let Δt be the time step. Assume that at time $t = n\Delta t$ f is given by a series

(3) $$f(x,u,u_r,n\Delta t) = \pi^{-1}(u_x^n)^{-2} \sum_{i=0}^{L_1} \sum_{j=0}^{L_2} a_{ij}(x,t) \cdot$$

$$\cdot H_i((u-v_x^n)/u_x^n) H_j(u_r/u_x^n) \exp(-((u-v_x^n)^2 + u_r^2)/(u_x^n)^2)$$

where v_x^n is the center of the expansion and u_x^n is its scale. The subscript x in u_x^n and v_x^n indicates that both parameters are allowed to vary with x, and it is assumed that $a_{ij} = 0$ for $i \geq L_1$, $j \geq L_2$. Appropriate v_x, u_x, L_1, L_2 will be determined below.

It is adequate to evaluate $a_{ij}(x)$ at the points $x = k\Delta x$, k integer, Δx a spatial increment, and we can write

$$a_{ijk}^n = a_{ij}(k\Delta x, n\Delta t) .$$

The density at time $n\Delta t$ is

$$\rho^n(x) = a_{00}(x, n\Delta t) = a_{00}^n(x) ,$$

the temperature is

$$T^n(x) = (u_x^n)^2((3/2)a_{00}^n + 2^{-1/2}a_{20}^n + 2^{1/2}a_{02}^n)/3R\rho^n \tag{4}$$

and the mean velocity is

$$\bar{u}^n = v_x^n + 2^{-1/2}u_x^n a_{10}^n/\rho^n . \tag{5}$$

Our aim is to obtain $f(x,u,u_r,(n+1)\Delta t)$ as a series of the form (3), but possibly with a new scale u_x^{n+1} and a new center v_x^{n+1} . To achieve this aim we evaluate the values f_{ijk}^{n+1} of $f(x,u,u_r,(n+1)\Delta t)$ at the points $x_k = k\Delta x$, $u_i = v_x^{n+1} + u_x^{n+1}\xi_i$, $u_{r,j} = u_x^{n+1}\xi_j$, where ξ_i, ξ_j are roots of $H_N(u) = 0$. The value of N remains to be chosen.

The algorithm for evaluating f_{ijk}^{n+1} will be described below.

Given $f(x,u,u_r,(n+1)\Delta t)$, the coefficients a_{ij} are defined by

$$(6)\quad a_{ij}^{n+1}(x) = \pi^{-1}(u_x^{n+1})^{-2} \iint f(x,u,u_r,(n+1)\Delta t)H_i((u-v_x^{n+1})/u_x^{n+1})\cdot$$

$$\cdot H_j(u_r/u_x^{n+1})du\,du_r$$

$$= \pi^{-1}(u_x^{n+1})^{-2} \iint f\ H_i((u-v^{n+1})/u_x^{n+1})H_j(u_r/u_x^{n+1})\cdot$$

$$\cdot\ \exp((u-v^{n+1})/u_x^{n+1})^2\ \exp(u_r/u_x^{n+1})^2\ .$$

$$\cdot\exp(-(u-v^{n+1})/u_x^{n+1})^2\ \exp\ -\ (u_r/u_x^{n+1})^2\ du\,du_r\ .$$

An obvious change of variables reduces the last integral to the form

$$\iint g(u,u_r)e^{-u^2}e^{-u_r^2}\ du\,du_r$$

which can be evaluated by Gauss-Hermite quadrature (see [15]), i.e. using a formula

$$(7)\quad \iint g(u,u_r)e^{-u^2}e^{-u_r^2}du\,du_r = \sum_{i=0}^{N}\sum_{j=0}^{N} g(\xi_i,\xi_j)w_iw_j\ ,$$

where ξ_i, ξ_j are the roots of $H_N(u) = 0$ and w_i, w_j are appropriate weights. Because of the choice of quadrature points, g is already

known at the appropriate points (ξ_i,ξ_j). Formula (7) is exact if $f(x,u,u_r)$ has a Hermite expansion of the form (3) with $L_1 = N$, $L_2 = N$ (see [11]). Thus as L_1, L_2 are increased, N should be increased.

There remains only the task of deciding how to evaluate f^{n+1}_{ijk}. This is done in the present paper by an explicit formula of the form

$$(8) \qquad f^{n+1}_{ijk} = A\, f^{n}_{ijk} + \Delta t\, Q(f,f)$$

where A is a linear operator such that $f^{n+1} - Af^{n}$ approximates $\frac{\partial f}{\partial t} - u\,\frac{\partial f}{\partial x}$, and $Q(f,f)$ is an approximation to the collision integral. We use

$$Af_{i,j,k} = (1-u\tfrac{\Delta t}{\Delta x})f_{i,j,k} + u\,\frac{\Delta t}{\Delta x}\, f_{i,j,k+s(u)}$$

where $s(u) = 1$ if $u < 0$ and $s(u) = -1$ if $u > 0$. This A has an obvious inituitive appeal, but since it is only of first order accuracy a more accurate approximation may be in order in future work. For stability we must of course have

$$(9) \qquad \frac{\Delta t}{\Delta x}\,\max_i |u_i| < 1.$$

In the evaluation of the collision integral, the crucial fact is that f^n is given as a continuous function of u and u_r, and thus no further interpolation problems will arise. Using the conservation of energy and momentum, the representation (3) of f, and an

obvious charge of variables, the collision term at $(x,u_i,u_{r,j})$ can be reduced to the form

$$C(x,u_i,u_{r,j}) \int_{-1}^{+1} d\phi \int_{-1}^{+1} d\chi \int_{-\infty}^{+\infty} du' \int_{-\infty}^{+\infty} du_r' \, G(\theta,\chi,u',u_r') e^{-u'^2} e^{-u_r'^2}$$

where C is a constant which depends on x, u_i, $u_{r,j}$. This integral can be approximated by the mixed Gauss and Gauss-Hermite quadrature formula

$$\sum_{i=0}^{N_1} \sum_{j=0}^{N_2} \sum_{k=0}^{N_3} \sum_{\ell=0}^{N_4} G(\theta_i,\chi_j,u_k',u_{r,\ell}') W_i W_j w_k w_\ell \tag{10}$$

where the θ_i, χ_j are the roots of the Legendre polynomials of degree N_1, N_2; u_k', $u_{r,\ell}'$ are the roots of $H_{N_3}(u) = 0$ and $H_{N_4}(u) = 0$, and W_i, W_j, w_k, w_ℓ are the quadrature weights, see [15].

Our method can thus be summarized as follows: At the beginning of each step, the solution f is given as a Hermite series of the form (3); f at the next level is evaluated at appropriate points using a difference scheme for the linear terms and weighted Gaussian quadrature for the collision term. The new values are then synthesized into a Hermite series. A variant of this method, using Monte-Carlo quadrature rather than Gaussian quadrature for the collision term, was presented in [3].

The fundamental difference between our approach and Hermite expansion methods such as Grad's thirteen moment approximation [6] lies in the fact that the number of Hermite polynomials used is not fixed in advance but depends on the course of the computation.

In particular, the adequacy of the representation at each step can be checked by using one more term and verifying that its effect is small.

In order to apply the algorithm just described, one needs an initial function f^0, as well as boundary conditions on f^n. Care must be exercised when the boundary conditions are imposed: $f(u,u_r)$ at a boundary may be imposed only for values of u, u_r such that the vector (u,u_r) points from the boundary into the gas. The distribution of the velocities of the particles coming from the fluid and hitting the boundary depends on the flow and cannot be imposed arbitrarily (see[7]). If this obvious condition is not respected, numerical instability will result.

Application to a shock problem. Consider a gas of elastic spheres flowing in $-\infty \leq x \leq +\infty$, with

(11) $$f(-\infty,u,u_r) = \rho_1\pi^{-1}U_1^{-2}\exp(-((u-v_1)^2+u_r^2)/U_1^2)$$

(12) $$f(+\infty,u,u_r) = \rho_2\pi^{-1}U_2^{-2}\exp(-((u-v_2)^2+u_r^2)/U_2^2) \;;$$

clearly $\bar{u}(-\infty) = v_1$, $\bar{u}(+\infty) = v_2$.

The Mach number M is defined by

(13) $$M = \sqrt{6/5}\; v_1/U_1 \;.$$

If $M > 0$ a shock will develop. There may be a steady shock if the following conservation laws are satisfied

(14) $$\rho_1 v_1 = \rho_2 v_2$$

(15) $$\rho_1 (v_1^2 + \tfrac{1}{2} U_1^2) = \rho_2 (v_2^2 + \tfrac{1}{2} U_2^2)$$

(16) $$\rho_1 v_1 (v_1^2 + \tfrac{5}{2} U_1^2) = \rho_2 v_2 (v_2^2 + \tfrac{5}{2} U_1^2) ,$$

where it is assumed that the ratio of specific heats is

$$\gamma = 5/3 .$$

From equation (14), (15), (16) we may deduce

(17) $$(U_2/U_1)^2 = (M^2+3)(5M^2-1)/16M^2$$

(18) $$(v_2/v_1) = (M^2+3)/4M^2$$

An important parameter in the shock problem is the shock thickness, conventionally (and awkwardly) defined by

(19) $$X = \frac{v_2 - v_1}{\max\limits_{x} |\frac{d\bar{u}}{dx}|} .$$

We pick $\rho_1 = 1$, $v_1 = 1$. Given M, equation (13), (17), (18) yield U_1, ρ_2, v_2, U_2. We call the left hand end of the shock the upstream side; we thus chose the units so that the mean free path upstream is one. In those units X is the ratio of the shock strength to the upstream mean free path; several authors have studied X^{-1} as a function of the Mach number M.

For practical reasons we replace the region $-\infty \leq x \leq +\infty$ by the region $-a \leq x \leq a$, where a is chosen large enough so that any further increase in a will have no noticable effect on the shock. At $x = a$ we impose the boundary condition

$$f(a,u,u_r) = \rho_2 \pi^{-1} U_2^{-2} \exp(-((u-v_2)^2 + u_r^2)/U_2^2) \text{ for } u + v_2 < 0$$

and at $x = -a$ we impose the condition

$$f(-a,u,u_r) = \pi^{-1} U_1^{-2} \exp(-((u-1)^2 + u_r^2)/U_1^2) \text{ for } u + 1 > 0 \ .$$

We divide $[-a,+a]$ into $K - 1$ segments, with a spatial increment

$$\Delta x = 2a/K \ .$$

Our aim it to obtain the steady shock profile as the limit, when the time tends to infinity, of an unsteady flow starting from an initial function $f^0 = f(x,u,u_r,0)$. This initial function should be chosen so that the steady limit is achieved as fast as possible. We first tried initial function f^0 resulting from an approximate

solution of the Boltzmann equation, in particular we tried the solution of the Mott-Smith u^2 theory [12]. This turned out to be a very poor choice. It is clear that the convergence to the steady limit is inherently slow-if we use K points across the shock, and if the stability condition (9) is respected, it takes at least $\dot{K}$ steps for the fastest particles to cross the shock. If f^0 is the Mott-Smith solution, the initial values assumed by $\frac{\partial f}{\partial t}$ are very small, and the relaxation to equilibrium takes an extremely long time, (showing, by the way, that the Mott-Smith solution is not a very good approximation to the real f). In addition, some odd effects appear: at low M the Mott-Smith theory overestimates the shock width, yet with Mott-Smith initial data the shock at first appears to widen; this effect can also be observed in the work of Haviland [9].

After considerable experimentation, it was found that an appropriate f^0 is the one which corresponds to a shock of zero width

$$f(x,u,u_r,0) = \begin{cases} f(-\infty,u,u_r) & \text{for } x \leq 0 . \\ f(+\infty,u,u_r) & \text{for } x > 0 . \end{cases} \tag{20}$$

The initial f given by (20) is particularly appropriate when one tries to determine the shock width X as defined by (19). X is a local property of the shock center, and with the data (20) X approaches equilibrium values long before $\frac{\partial f}{\partial t}$ becomes close to zero.

It is worth noting that from the numerical point of view the

determination of the shock width X is a comparatively difficult undertaking, since it requires high accuracy in the region of fastest variation of f. In a variety of other problems, e.g. problems involving the interaction of a shock with a boundary, the choice of initial data is less critical and the computation is easier to carry out.

We now apply the method outlined earlier to the study of the shock wave. There is a considerable number of numerical parameters to be chosen: the centers v_x^n and scales u_x^n of the expansion (3), as well as the number $(L_1+1)(L_2+1)$ on nonzero terms; the size 2a of the region of integration, the spatial increment Δx, the time step Δt, the number of quadrature points $N_1N_2N_3N_4$ in each evaluation of the collision integral and the number N^2 of points at which f^{n+1} is evaluated given f^n.

We choose v_x^n and u_x^n as follows:

(21)
$$
\begin{aligned}
v_x^{n+1} &= \bar{u}^n(x) \\
u_x^{n+1} &= \sqrt{2RT^n(x)}
\end{aligned}
$$

i.e. we expand at each step around the mean velocity at the preceding step and using a scale determined by the temperature at the preceding step. $\bar{u}^n$, T^n, are given by (4) and (5). This choice is not the only reasonable one, and will be further discussed below.

The width 2a of the region of integration was chosen by trial and error, generally around 25 mean free paths. Δx is chosen small

enough so that any further decrease in Δx will not affect the outcome of the calculation. We proceed as follows: We evaluate $\frac{d\bar{u}}{dx}$ which enters the definition (19) of X using both the formula

$$\frac{d\bar{u}}{dx} \cong \frac{\bar{u}_{k+1}-\bar{u}_{k-1}}{2\Delta x} \tag{22}$$

and

$$\frac{d\bar{u}}{dx} \cong \frac{\bar{u}_{k+1}-\bar{u}_{k}}{\Delta x} \quad . \tag{23}$$

which are of different orders in Δx; when they are in substantial agreement Δx can be considered sufficiently small. It was found that Δx of order 1 (i.e. one mean free path) is generally adequate; under these circumstances, X evaluated with the use of (23) is a more reliable estimate of the true X, since X is a local property of the shock center and an estimate using (23) depends on the values of f in a smaller neighborhood.

The stability condition (9) gives a good estimate of the appropriate value of Δt. We usually choose Δt to be 0.8 times the maximum value allowed by (9). Higher values of Δt may give rise to instability in the presence of temperature overshoots while lower values lengthen the computation without increasing its accuracy. At this point we have to introduce an additional numerical parameter. It is readily seen that the stability of the scheme $f^{n+1} = Af^n$ would imply the stability of the complete scheme (8) if only the integrand on the right hand side had compact support. This last condition is not satisfied, but f does decrease rapidly with increasing $|u|$, $|u_r|$, so that one might assume that condition

(9) is sufficient for stability. Numerical experimentation shows this to be the case whenever $L_1 \leq 3$ and $L_2 \leq 3$. However, when L_1 or L_2 is larger, the range of u, u_r over which f is not negligible increases, and it is necessary to truncate the support of f. This can be done by setting $f^n = 0$ whenever $|u| \geq v_x^n + \xi\lambda u_x^n, |u_r| \geq \lambda\xi u_x^n$, where u_x^n is the scale of the expansion, v_x^n its center, ξ is the largest root of $H_N(u) = 0$ and λ is a constant larger than 1. When $\lambda > 1$ such truncation leads to no decrease in accuracy, since the expansion in Hermite polynomials is not uniformly valid in u, u_r. We generally chose $\lambda \sim 1.1$.

We generally took $L_1 = L_2$, equal to an integer L. Clearly we must have $L < N$; on the other hand if N were much larger than L, information would be generated and immediately discarded; so we generally chose $N = L + 1$ (L even) and $N = L + 2$ (L odd). The difference between the odd and even cases is due to programming consideration and is of no particular significance.

This leaves open the choice of L, the number of Hermite polynomials in each of the variables u, u_r. It would be natural to choose L so large that $a_{ij} \cong 0$ for either i or j larger than L. It turns out however that a_{ij} decays much more slowly with i and j than expected, but that the presence of the higher terms in the expansion affects but little the computed values of X and of the density, mean velocity and temperature. For example, at Mach number $M = 1.6$, a_{04} near the center of the shock tends to the steady value $a_{04} \cong -.4$, yet within computational error there is no difference between the value of X computed with $L = 4$ and the value computed

with $L = 3$, i.e. neglecting a_{04}. It does appear therefore that the lower moments of f are almost independent of the higher moments, a result both surprising and natural. It also appears that the assumption underlying Grad's thirteen moment approximation [5], namely that the coefficients of the Hermite polynomials of degree greater than 3 are small, is not correct in itself but could lead to correct answers. We made runs with both $L = 3$ and $L = 4$. It must be added that although the values of X do not seem to depend on L provided $L \geq 3$, when $M < 2$, the initial rate of change in X does depend on L. This is probably of no physical significance, since the initial data are wholly unrealistic. The relationship between our method and Grad's will be the object of further investigation elsewhere. It should be noted that when $L = 3$ our f is represented by 8 coefficients a_{ij}, taking into account the fact that by symmetry $a_{ij} = 0$ for odd j; when $L = 4$ our f is represented by 15 functions. This compares with 5 functions for the one-dimensional case of Grad's expansion.

N_1, N_2, N_3, N_4 are also chosen by trial and error. We must have $N_3 > L/2$, $N_4 > L/2$, so that the highest moments of f used enter the collision integral. It was generally found that with $L = 3$ or $L = 4$, the choice $N_1 = N_2 = N_3 = N_4 = 3$, i.e. 81 integration points for every evaluation of the collision integral, was quite adequate. The fact that such low values are adequate is testimony both to the power of Gaussian quadrature and to the aptness of the representation (3).

The existence of conservation laws affords a natural check on

accuracy, since no exact conservation is built into our scheme. With the initial data (20), and with a large enough, the mass, momentum and energy in the shock region are constant. The magnitude of the numerically induced variations in, say, the mass provides a reasonable indication of the accuracy of the computation.

In tables I and II we display the relaxation from the initial data (20). In table I the mean velocity is tabulated as a function of x for low values of $t/\Delta t$ and at Mach number 2; this should give a qualitative picture of the behavior of the numerical process. In table II the instantaneous value X^{-1} of the reciprocal of the shock width, the maximum of $|\frac{\partial \bar{u}}{\partial t}|$, the location of that maximum, and the computed total mass Q in the shock region, are tabulated as functions of $t/\Delta t$ for $M = 2$. It is seen that $|\frac{\partial u}{\partial t}|$ does not decay to zero fast, if at all, and that X^{-1} oscillates. In each run we therefore estimated the range of values assumed by X^{-1}, defined as the range between the last maximum and last minimum of X^{-1}. It is not clear whether the oscillations ever die out. They are amplified if the width of the region of integration 2a is chosen too small, but they can no longer be decreased by a further increase in 2a. The location of the maximum of $|\frac{\partial u}{\partial t}|$ recedes in time, showing that upstream convergence is slower than downstream. Similar observations were made by Haviland [9]. Q, the total mass, is evaluated by

$$Q = \sum_{i=0}^{K} \rho(i\Delta x)\Delta x \quad , \tag{24}$$

Table I

$\bar{u}$ as a function of x and t.

M = 2, Δt = .413, Δx = 1.5

x	$t/\Delta t$ = 1	$t/\Delta t$ = 4	$t/\Delta t$ = 8
-12.75	1.000	1.000	1.000
-11.25	1.000	1.000	1.000
-9.75	1.000	1.000	1.000
-8.25	1.000	1.000	1.000
-6.75	1.000	1.000	.999
-5.25	1.000	.999	.999
-3.75	1.000	.999	.998
-2.25	1.000	.995	.961
-0.75	.979	.873	.726
0.75	.461	.499	.535
2.25	.437	.458	.478
3.75	.437	.444	.460
5.25	.437	.438	.451
6.75	.437	.437	.444
8.25	.437	.437	.440
9.75	.437	.437	.438
11.25	.437	.437	.437
12.75	.437	.437	.437

Table II

Relaxation to a steady shock

$M = 2,\ \Delta t = .413,\ \Delta x = 1.5$

$t/\Delta t$	X^{-1}	$\max\left\lvert\frac{\partial u}{\partial t}\right\rvert$	location of $\max\left\lvert\frac{\partial u}{\partial t}\right\rvert$	Q
1	.61	.056	+ .75	44.31
2	.56	.069	- .75	44.29
3	.50	.088	- .75	44.29
4	.44	.097	- .75	44.31
5	.38	.099	- .75	44.33
6	.32	.096	- .75	44.35
7	.27	.087	- .75	44.37
8	.27	.072	- .75	44.40
9	.28	.057	- .75	44.45
10	.28	.043	-2.25	44.48
11	.28	.049	-2.25	44.51
12	.26	.055	-2.25	44.54
40	.20	.056	-6.75	44.30
41	.19	.060	-6.75	44.27
42	.19	.061	-6.75	44.25
43	.21	.060	-6.75	44.22
44	.23	.057	-6.75	44.19
45	.24	.052	-6.75	44.17
46	.25	.045	-6.75	44.15
47	.24	.045	-8.25	44.13
48	.23	.051	-8.25	44.11
49	.21	.056	-8.25	44.10

Table III

Structure of a shock

$M = 2$, $t = 9.5192$

x	$\bar{u}$	ρ	T	H/ρ
-12.75	1.000	1.000	.300	-1.398
-11.25	.999	1.000	.300	-1.398
- 9.75	.999	1.000	.300	-1.398
- 8.25	.999	1.000	.300	-1.398
- 6.75	.998	1.001	.301	-1.403
- 5.25	.977	1.020	.316	-1.467
- 3.75	.831	1.165	.431	-1.787
- 2.25	.670	1.371	.622	-2.270
- .75	.660	1.392	.680	-2.479
+ .75	.616	1.587	.660	-2.299
2.25	.546	1.937	.628	-2.011
3.75	.501	2.213	.609	-1.802
5.25	.485	2.342	.602	-1.712
6.75	.479	2.381	.601	-1.692
8.25	.477	2.379	.605	-1.708
9.75	.472	2.367	.612	-1.743
11.25	.467	2.349	.619	-1.788
12.75	.437	2.285	.623	-1.883

Table IV

Coefficients a_{ij}

Mach number = 1.6, x = 1.5, t = 10.76

	i = 0	i = 1	i = 2	i = 3	i = 4
j = 0	1.447	-.0002	.48	-.04	-.38
j = 2	.24	-.19	-.005	.05	.01
j = 4	.11	.08	.004	-.02	-.006

$a_{ij} = 0$ for odd j.

Table V

Reciprocal shock width X^{-1} as a function of Mach number M.

	L = 3	L = 4	Gilbarg and Paolucci	Mott-Smith	Ziering et al.
M = 1.4	.12 to .13		.136	.116	.181
M = 1.6	.22 to .24	.22 to .24		.164	.238
			.222		
M = 1.8	.18 to .21			.205	.284
M = 2.0	.19 to .25	.23 to .29	.381	.235	.324

it is seen that Q varies little; whatever variations there are can be ascribed to the inaccuracy of the formula (24).

In table III we display the structure of a typical shock. The mean velocity $\bar{u}$, density ρ, temperature T, and Boltzmann H divided by ρ, are given as functions of x, for M = 2 and t = 9.5192. The familiar features of the shock appear: $\bar{u}$ and ρ vary in a monotone fashion; T exhibits an overshoot, see [16]; H/ρ, which is determined up to an additive constant, displays a dip. H is evaluated from f using, as usual, Gauss-Hermite quadrature.

In table IV we present the coefficients a_{ij} for x = 1.5, M = 1.6, t = 10.76. The purpose of the table is to show that a_{40} at that point is not small.

Some of the more interesting results are grouped on table V, where the ranges of oscillation of X^{-1} for Mach numbers 1.4, 1.6, 1.8, and 2.0 are given, with both L = 3 and L = 4, and compared with the values of X^{-1} computed by Gilbarg and Paolucci using the Navier-Stokes equations, and by Mott-Smith and Ziering et al using their respective theories. As expected, at M = 1.4 the computed X^{-1} is very close to the Navier-Stokes result. At M = 1.6, where the result is seen to be independent of $L \geq 3$, the shock is thinner than the Navier-Stokes shock, with X^{-1} close to the value given by Ziering et al. Although our method is clearly inspired by Grad's work, and although some of Grad's ideas are resoundingly vindicated, the numerical results do not agree with Grad's, whose shocks are always thicker than the Navier-Stokes shocks. It seems that five moments are just one or two short of giving an accurate description of the shock.

Between $M = 1.6$ and $M = 1.8$ there seems to be a change of regime; suggestively this occurs in the region where Grad's approximation breaks down. Above $M = 1.8$ the results seem to agree with the Mott-Smith predictions.

Comparison of these results with available Monte-Carlo results is difficult, since the Monte-Carlo calculations in the literature cover time spans too short to be of any significance. The results contradict the conclusions of Bird [1], whose shocks are always thicker than the Navier-Stokes shocks, and they are in some qualitative agreement with the conclusion of Haviland [9], but one may wonder whether this is more than coincidence.

Generalizations and comments. It is quite clear that the procedure of the preceding section will break down, for a fixed number of terms in the Hermite expansion, whenever the Mach number is large enough; certainly by the time all the velocities $u = v_x + u_x \xi_i$, $i = 0,\ldots,N$, ξ_i roots of $H_N(u) = 0$, are of the same sign. With u_x, v_x given by (21) and $N = 5$ this breakdown occurs just above Mach number 2. One could keep increasing the number of polynomials as M increases; it is more reasonable to systematize the Mott-Smith and Ziering et al procedures by representing f as a sum of two series of the form (3), with scales and centers determined respectively be the conditions upstream and downstream from the shock.

Other changes in the scaling (21) may be justified: for example,

it is probably beneficial to introduce two distinct scalings for the variables u and u_r.

Another modification our basic method was explained in [3]: the evaluation of the collision integral may be performed by Monte-Carlo quadrature, with the possible help of the variance reduction technique introduced in [3]. This should be particularly effective close to equilibrium when the integrand of the collision term is small, provided this term is not separated into gain and loss terms, as was done by Nordsieck [13].

The methods of this paper are readily generalized to problems in more dimensions and with other types of interparticle force.

Bibliography

[1] G.A. Bird, Shock Wave Structure in a Rigid Sphere Gas, Rarefied Gas Dynamics, Suppl 3, Vol. I (1965).

[2] S. Chapman and T.G. Cowling, The Mathematical Theory of Non-Uniform Gases, Cambridge University Press (1958).

[3] A.J. Chorin, Hermite Expansions in Monte-Carlo Computation, to appear in J. Comput. Physics.

[4] D. Gilbarg and D. Paolucci, The Structure of Shock Waves in the Continuum Theory of Fluids, J. Rat. Mech. Anal., 2, 617.

[5] H. Grad, On the Kinetic Theory of Rarefied Gases, Comm. Pure Appl. Math., 2, 311 (1949).

[6] H. Grad, The Profile of a Steady Plane Shock Wave, Comm. Pure Appl. Math., 5, 257, (1952).

[7] H. Grad, Principles of the Kinetic Theory of Gases, Handbuch der Physik, Vol. XII, Springer-Verlag (1958).

[8] J.K. Haviland, Determination of Shock-Wave Thickness by the Monte-Carlo Method, Proc. 3rd Symp. Rarefied Gas Dynamics, Academic Press (1963).

[9] J.K. Haviland, The Solution of Two Molecular Flow Problems by the Monte-Carlo Method, Methods in Computational Physics, Vol. 4, p. 109 (1965).

[10] S. Kaczmarz and H. Steinhaus, Theorie der Orthogonalreitan, Warsaw (1935).

[11] C. Lanczos, Applied Analysis, Prentice Hall (1956).

[12] H.M. Mott-Smith, The Solution of the Boltzmann Equation for a Shock Wave, Physical Review, 82, 885 (1951).

[13] A. Nordsieck and B.L. Hicks, Monte-Carlo Evaluation of the Boltzmann Collision Integral, Proc. 5th Symp. Rarefied Gas Dynamics, Academic Press (1967).

[14] A. Sommerfeld, Thermodynamics and Statistical Mechanics, Academic Press (1964).

[15] A.M. Stroud and D.Secrest, Gaussian Quadrature Formulas, Prentice Hall (1966).

[16] S.M. Yen, Temperature Overshoots in Shock Waves, Phys. Fluids, 9, 1417 (1966).

[17] S. Ziering, F. Ek and P. Koch, Two-Fluid Models For the Structure of Neutral Shock Waves, Phys. Fluids, 4, 975 (1961).

Appendix

In the appendix we list the program used to obtain the results given above. It should be borne in mind that this program was written, not in view of minimizing computing time per run, but rather so that experimentation and change are as easy as possible. Obvious ways in which the running time could be shortened are: better exploitation of the symmetry $f(x,u,u_r) = f(x,u,-u_r)$; use of the recursion relation between Hermite polynomials, and restriction of the calculation to regions in space where substantial changes are occuring. As it stands, the program performs one time step in approximately a minute (with $K = 18$, $L = 4$, $N_1 = N_2 = N_3 = N_4 = 3$, $N = 5$) on the CDC 6600 computer.

```
      PROGRAM COLLIDE (OUTPUT)
      COMMON CA,CB,CC,CD,CE,PI,TPI,TIME,C1,C2,C3,C4,C5,
     1 C6,C7,C8,C9,C10,ROOT(10),WEIT(10       )
     3 , POOT(10),PEIT(10)
      DIMENSION UU(5,5,40),VV(5,5,40),D(5,5,40)
     1 ,TT(40),UX(40),BOL(40),TRL(40)
     1 ,URL(40)
              IPR=-1
C            IPR DETERMINES WHETHER TO PRINT OR NOT
C     TT IS THE SQUARE OF THE THERMAL VELOCITY
C             TRL=RHO*TT=D(1,1,IC)*TT(IC)
C     RATIO OF SPECIFIC HEATS GAMMA=5/3
      M=3
      M=5
      MM=3
      MMM=3
           CALL HERMY
     1 (M,MMM)
      PIH=PI/2.
          C0=0.
          PRINT 9020,C0,C1,C2,C3,C4
      NCOUN=5
      ICOUN=1
      EPS=0.
      TIME=0.
      NC=18
          NHALF=NC/2
            NHALFP=NHALF+1
              NCM=NC-1
           CNC=NC-1
      DX=1.5
      XXX=DX*CNC
C     LA,LB NUMBERS OF POLYNOMIALS
      LA=4
      LB=LA
C     RR UNIVERSAL GAS CONSTANT
        RR=0.5
      XMACH=2.
         XMM=XMACH*XMACH
           U1=1.
          R1=1.
          CC1=U1*U1*(6./5.)/XMM
            U2=U1*((XMM+3.)/(4.*XMM))
          CC2=CC1*((XMM+3.)*(5.*XMM-1.)/(16.*XMM))
           R2=R1*U1/U2
      E1=CC1*R1
      E2=CC2*R2
         PRINT 9025
             PRINT 9017, XMACH,U1,R1,CC1,E1,U2,R2,CC2,E2
```

```
             BAM=R1*(U1*U1+0.5*CC1)
            BAMM=R2*(U2*U2+0.5*CC2)
              PRINT 9020, BAM,BAMM
C
C
C           UXM IS THE LARGEST MEAN VELOCITY
      UXM=1.2
      UMAX=ABS(ROOT(M))
     1*1.2
     1 +UXM
            DT=DX/UMAX
           PRINT 9023,DT,DX
           XLAM=DT/DX
      COFB=DT/(SQRT(2.)*PI)
C     DISTANCE MEASURED IN MEAN FREE PATHS,
C     TIME IN COLLISION TIMES, VEL IN TH VEL UNITS
C
C
C
C
C
C           SET ALL TO ZERO
             DO 120 IC=1,NC
      BOL(IC)=0.
            DO 120 IA=1,5
              DO 120 IB=1,5
        D(IA,IB,IC)=0.
  120 CONTINUE
C       SETTING INITIAL DATA
      IC=1
      IC=NC
              DO 121 IC=1,NHALF
      TRL(IC)=E1
              UX(IC)=U1
      URL(IC)=UX(IC)
            TT(IC)=CC1
            D(1,1,IC)=R1
  121 CONTINUE
              DO 122 IC=NHALFP,NC
             UX(IC)=U2
      TRL(IC)=E2
      URL(IC)=UX(IC)
            TT(IC)=CC2
             D(1,1,IC)=R2
  122 CONTINUE
C
C
            DO 731 IC=1,NC
```

```
          CD=2.*RR*TT(IC)
          CE=PI*CD
       DO 500 I=1,M
          DO 500 J=1,M
         QX=ROOT(I)
          QY=ROOT(J)
         GLOG=0.
         DO 501 IA=1,LA
         IAM=IA-1
         DO 501 IB=1,LB
             IBM=IB-1
         GLOG=QLOG+D(IA,IB,IC)
    1*H(IAM,QX)
    2 *H(IBM,QY)
 501 CONTINUE
        UU(I,J,IC)=GLOG*EXP(-QX*QX-QY*QY)/CE
 500 CONTINUE
     IF(IPR.LE.0) GO TO 1806
     PRINT 9004
     PRINT 9022,IC
     PRINT 9004
     DO 790 J=1,M
 790 PRINT 9001,(UU(I,J,IC),I=1,M)
1806 CONTINUE
 731 CONTINUE
C
C
C
C
C
C
C          TIME STEP        TIME STEP    TIME STEP
      NSTP=12
      NSTP=10
       NSTP=20
      NSTP=35
      NSTP=50
          DO 300 ISTP=1,NSTP
      PRINT 9007,ISTP
      TIME=TIME+DT
           PRINT 9026, TIME
C
C          A POINT IN SPACE
C
C
      DO 700 IC=2,NCM
          SIG=RR
            CD=2.*RR*TT(IC)
```

```
          CC=SQRT(CD)
      CE=PI*CD
            CA=CC
         CB=CC*CD
        CCT=CC
      RSCL=SQRT(TPL(IC)/(D(1,1,IC)*TT(IC)))
C
C          CONVECTION TERMS
C
C
       DO 730 I=1,M
          IP=IC+1
      VELC=ROOT(I)*RSCL
     1 +C1*D(2,1,IC)
     1 /D(1,1,IC)
      VEL=VELC*CC+UX(IC)
            IF(VEL.GE.0.)    IP=IC-1
           IF(IP.LT.1)    IP=1
             IF(IP.GT.NC)    IP=NC
          CP=ABS(VEL*DT)/DX
          CENT=1.-CP
           RATP=TT(IC)/TT(IP)
         CEP=CE/RATP
        RATP=SQRT(RATP)
      VELP=VELC+(UX(IC)-UX(IP))/CC
           VELP=VELP*RATP
      DO 730 J=1,M
           VELRC=ROOT(J)
     1*RSCL
          VELR=VELRC
         VELRP=ROOT(J)*RATP
     1*RSCL
          CAME=0.
           DO 710 IA=1,LA
        IAM=IA-1
        DO 710 IB=1,LB
     1    ,2
         IBM=IB-1
        CAME=CAME
     1   +CENT*D(IA,IB,IC)*H(IAM,VELC)*H(IBM,VELRC)
     2  * EXP(-VELC*VELC-VELRC*VELRC)/CE
     1  +CP*D(IA,IB,IP)*H(IAM,VELP)*H(IBM,VELRP)
     2  *EXP(-VELP*VELP-VELRP*VELRP)/CEP
 710  CONTINUE
          VV(I,J,IC)=CAME
 730  CONTINUE
           IF(IPR.LE.0)  GO TO 1800
           PRINT 9022,IC
```

```
          PRINT 9004
         PRINT 9020,CA,CB,CC,CD,CE
          PRINT 9004
         DO 734 J=1,M
  734 PRINT 9001,(VV(I,J,IC),I=1,M)
 1800 CONTINUE
C
C
C
C
C
C
C
C                   VARIOUS POINTS IN U-SPACE
C
        DO 100 JQ=1,M
           DO 100 IQ=1,M
         RSLT=0.
          UAP=ROOT(IQ)
      1*RSCL
      1 +C1*D(2,1,IC)
      1 /D(1,1,IC)
          UBP=ROOT(JQ)
      1*RSCL
           CATA=0.
           DO 410  IA=1,LA
            IAM=IA-1
          DO 410 IB=1,LB
      1    ,2
             IBM=IB-1
           CATA=CATA+D(IA,IB,IC)
      1 *H(IAM,UAP)*H(IBM,UBP)
  410 CONTINUE
C     CONSTANTS       PI*PI/4    (FROM ANGLES)
C     *CD(TWO INTEGRATIONS)  CC (FROM VE ) / PI*PI*CD*CD
C       FROM NORMALIZATION OF FF, /2 FROM
C      RESULTING IN DIVISION BY
C       8*CC
      Q=EXP(-UAP*UAP-UBP*UBP)
      RUG=8.*CC
      Q=Q/RUG
C
      CALL HERMY(MM,MMM)
              DO 10 JA=1,MM
             DO 10 JB=1,MM
              DO 10 JC=1,MMM
      DO 10 JD=1,MMM
C
```

```
C
C
C          NEW COLLISION
              COLLA=ROOT(JA)
              COLLB=ROOT(JB)
       ANGLE=PIH+PIH*ROOT(JC)
          BA=COS(ANGLE)
        BB=SIN(ANGLE)
        AZ=PIH+PIH*ROOT(JD)
        CAZ=COS(AZ)
       SAZ=SIN(AZ)
       BBC=BB*CAZ
       BBS=BB*SAZ
       VE=(COLLA-UAP)*BA
     1 +(COLLB*CAZ-UBP)*BBC
     1 +COLLB*SAZ*BBS
          UA=UAP+VE*BA
       UB=UBP+VE*BBC
       UC=UBP+VE*BBS
       UB=SQRT(UB*UB+UC*UC)
           GAUSA=COLLA-VE*BA
       GAUSB=COLLB-VE*BBC
       GAUSC=COLLB-VE*BBS
       GAUSB=SQRT(GAUSB*GAUSB+GAUSC*GAUSC)
           DATA=0.
       DO 205 IA=1,LA
           DO 205 IB=1,LB
     1    ,2
         IBM=IB-1
       IAM=IA-1
         DATA=DATA+D(IA,IB,IC)
     1 *H(IAM,UA)
     2 *H(IBM,UB)
 205 CONTINUE
            PATA=0.
       DO 210 IA=1,LA
       IAM=IA-1
          DO 210 IB=1,LB
     1    ,2
         IBM=IB-1
         PATA=PATA+D(IA,IB,IC)
     1 *H(IAM,GAUSA)
     1 *H(IBM,GAUSB)
 210 CONTINUE
           QATA=0.
           DO 411 IA=1,LA
            IAM=IA-1
          DO 411 IB=1,LB
```

```
     1     ,2
              IBM=IB-1
           QATA=QATA+D(IA,IB,IC)
     1  *H(IAM,COLLA)*H(IBM,COLLB)
  411 CONTINUE
      AUX=(DATA*PATA-QATA*CATA)*ABS(VE)
     1  *SIN(ANGLE)
              RSLT=RSLT+AUX*WEIT(JA)*WEIT(JB)*PEIT(JC)
     1  *PEIT(JD)
   10 CONTINUE
      CALL HERMY(M,0)
   30 CONTINUE
             IF(RSLT.LE.0.) RSLT=0.
         UU(IQ,JQ,IC)=RSLT*COFB
     1  *Q
     1  +VV(IQ,JQ,IC)
  100 CONTINUE
              IF(IPR.LE.0)    GO TO 1801
          PRINT 9016
         DO 302 J=1,M
          PRINT 9001,(UU(I,J,IC),I=1,M)
  302 CONTINUE
 1801 CONTINUE
  700 CONTINUE
C
C
C
C
C
C
C
C
C        RESCALING
      DO 1707 IC=2,NCM
      TT(IC)=TRL(IC)/D(1,1,IC)
      UX(IC)=URL(IC)
 1707 CONTINUE
C
C
C          ASSUMES RR*2.=1.
      ICOUN=ICOUN+1
      IF(ICOUN.GT.NCOUN)     ICOUN=1
C
C
C
C
               DO 1700 IC=1,NC
                CD=2.*RR*TT(IC)
```

```
      CC=SQRT(CD)
         CE=PI*CD
         CA=CC
       CB=CC*CD
C
C
C
         ANALYSIS OF DATA
      DO 177 IA=1,LA
         DO 177 IB=1,LB
            D(IA,IB,IC)=0.
177 CONTINUE
      DO 102   J=1,M
   DO 102 I=1,M
      WEIGH=WEIT(I)*WEIT(J)
 1 *CA
      UUU=UU(I,J,IC)
     ZA=ROOT(I)
      ZB=ROOT(J)
     AA=EXP(ZA*ZA+ZB*ZB)*UUU*WEIGH*CA
       DO 102 IA=1,LA
      DO 102 IB=1,LB
       IAM=IA-1
     IBM=IB-1
       D(IA,IB,IC)=D(IA,IB,IC)
 1 +AA*H(IAM,ZA)*H(IBM,ZB)
102 CONTINUE
C
C
C    ENTROPY
       RHO=D(1,1,IC)
        BOL(IC)=0.
   DO 750 I=1,M
   DO 750 J=1,M
   DO 750 K=1,M
   ZA=ROOT(I)
   ZB=ROOT(J)
   ZC=ROOT(K)
   ZR=SQRT(ZB*ZB+ZC*ZC)
   VAL=0.
   DO 751 IA=1,LA
   IAM=IA-1
   DO 751 IB=1,LB
  1 ,2
   IBM=IB-1
   VAL=VAL+D(IA,IB,IC)*H(IAM,ZA)*H(IBM,ZR)
751 CONTINUE
   VALE=EXP(-ZA*ZA-ZR*ZR)/CE
   VALE=VAL*VALE
```

```
      IF(VALE.LE.1.E-6) GO TO 750
      VAL=VAL*ALOG(VALE)
      BOL(IC)=BOL(IC)+VAL*WEIT(I)*WEIT(J)
     1*WEIT(K)
  750 CONTINUE
      COT=RHO*PI
     1 /CC
      BOL(IC)=BOL(IC)/COT
C
C
C          TEMPERATURE
      TREAL=CD*((3./2.)*D(1,1,IC)+SQRT(0.5)*D(3,1,IC)
     1 + SQRT(2.)*D(1,3,IC))
          TREAL=TREAL/3.
          TREAL=TREAL/RR
C         UPDATING
      URL(IC)=UX(IC)+C1*CC*D(2,1,IC)
     1 /D(1,1,IC)
            TRL(IC)=TREAL
      IF(ICOUN.NE.NCOUN)  GO TO 764
            PRINT 9022, IC
      PRINT 9004
      PRINT 9020, CA,CB,CC,CD,CE
      PRINT 9004
         PRINT 9004
C
       PRINT 9015
         DO 232 IB=1,LB
         PRINT 9020,(D(IA,IB,IC),IA=1,LA)
  232 CONTINUE
             PRINT 9004
C
  764 CONTINUE
 1700 CONTINUE
C
C
C
             PRINT 9004
C
C
C        CHECK ON CONSERVATION
             PRINT 9010
      SA=0.
      SB=0.
      SC=0.
      ABRA=1.
          DO 221 IC=1,NC
             QZ=UX(IC)
```

```
      QZ=URL(IC)
            QUQU=D(1.1,IC)
      QUP=0.5*TRL(IC)/QUQU
           QA=QUQU*QZ
          QB=QUQU*(QZ*QZ+QUP)
             QC=QUQU*QZ*(QZ*QZ+5.*QUP)
       SA=SA+QA
      SB=SB+QB
      SC=SC+D(1,1,IC)
            PRINT 9018, QA,QB,QC
  221 CONTINUE
      SA=SA*DX
      SB=SB*DX
      SC=SC*DX
      PRINT 9004
      PRINT 9004
      PRINT 9018,SA,SB,SC
      PRINT 9004
      PRINT 9004
  470 CONTINUE
C
C
C          SUMMING UP
C
            PRINT 9004
          PRINT 9008
        PRINT 9025
        DO 222 IC=1,NC
            STR=0.
      HI=TRL(IC)/D(1,1,IC)
      PRINT 9020,URL(IC),D(1,1,IC),FI,BOL(IC)
  222 CONTINUE
C
C
C
C
C     EXIT AND REFINEMENT
      ERR=0.
      DO 760 IC=1,NC
      CRT=ABS(URL(IC)-UX(IC))
      IF(ERR.LE.CRT)  ERR=CRT
      IF(ERR.LE.CRT)  ING=IC
  760 CONTINUE
        ERR=ERR/DT
      PRINT 9002,ERR
      PRINT 9006,ING
      IF(ERR-EPS)  761,762,762
  761 CONTINUE
```

```
      CALL EXIT
  762 CONTINUE
C
C
C     SHOCK THICKNESS
      TDX=2.*DX
      SLOPE=0.
      SLOPC=0.
      DO 1900 IC=2,NCM
      SLXC=ABS(URL(IC+1)-URL(IC-1))/TDX
      IF(SLOPC.LE.SLXC)  SLOPC=SLXC
      IOT=IC-1
      SLX=ABS(URL(IC)-URL(IC-1))/DX
      IF(SLOPE.LE.SLX)      SLOPE=SLX
       IF(SLOPE.LE.SLX)   ING=IC
 1900 CONTINUE
      SHW=(U1-U2)/SLOPE
      SHWC=(U1-U2)/SLOPC
      PRINT 90 28, SHW, SHWC
      G=1./SHW
      GG=1./SHWC
      PRINT 9003,G,GG
      PRINT 9006,ING
  300 CONTINUE
C             END OF TIME STEP
C
C
C
C
C              COMPUTATION OF NORMAL VELOCITY COMPONENT
          PRINT 9024
C     LOX MUST BE ODD
       LOX=21
      LOXH=LOX/2
      CLOXH=LOXH
      GAMUT=2.
      DN=GAMUT/CLOXH
      DO 770 IRT=1,3
      IF(IRT.EQ.1) IC=1
      IF(IRT.EQ.2) IC=NHALF
      IF(IRT.EQ.3) IC=NC
      PRINT 9004
      PRINT 9022,IC
      PRINT 9004
      CC=SQRT(TT(IC))
      CE=PI*TT(IC)
      DO 770 I=1,LOX
      IM=I-1
```

```
      CIM=IM
      PLC=-GAMUT+CIM*CN
      PLC=PLC-UX(IC)
      PLC=PLC/CC
      CHAT=0.
      DO 901 J=1,M
      PLCN=ROOT(J)
      CHIEN=0.
      DO 900 IA=1,LA
      IAM=IA-1
      DO 900 IB=1,LB
     1 ,2
      IBM=IB-1
      CHIEN=CHIEN
     1 +D(IA,IB,IC)*H(IAM,PLC)*H(IBM,PLCN)
  900 CONTINUE
      CHIEN=CHIEN*EXP(-PLC*PLC)
      CHIEN=CHIEN*CC/CE
      CHAT=CHAT+CHIEN*WEIT(J)
     1*PI
  901 CONTINUE
      PRINT 9020, CHAT
  770 CONTINUE
C
C
C
C
C
C
 9001 FORMAT (1X,6F11.7)
 9002 FORMAT(4H ERR, F11.7)
 9003 FORMAT(12H RECIPROCALS,2F11.7)
 9004 FORMAT(1X)
 9005 FORMAT(6H EXACT,F11.7)
 9006 FORMAT(1X,20I5)
 9007 FORMAT(/5H STEP,I5/)
 9008 FORMAT(14H SHOCK PROFILE)
 9009 FORMAT(5H TEMP, F11.7)
 9010 FORMAT(13H CONSERVATION)
 9011 FORMAT (5H RSLT,F11.7,3HVAR,F11.8)
 9012 FORMAT(6H POINT,I5)
 9013 FORMAT(16H IS THE SOL GOOD, F11.7)
 9015 FORMAT(8H COEFF D)
 9016 FORMAT(3H UU/)
 9017 FORMAT(8H MACH NO, F11.7/1X,4F11.7/1X,4F11.7/)
 9018 FORMAT(1X,3(F11.7,5X))
 9020 FORMAT(1X,5F11.7)
 9021 FORMAT(1X,4F11.7,2I3)
```

```
9022 FORMAT(6H SPACE,I3)
9023 FORMAT(6H DT DX, 2F11.7)
9024 FORMAT(14H N VEL PROFILE)
9025 FORMAT(1X,1HU,10X,1HR,10X,1HT,10X,3HBOL/)
9026 FORMAT(5H TIME, F11.7)
9027 FORMAT(14H NAVIER STOKES)
9028 FORMAT (/12H SHOCK WIDTH,2F11.7/)
     CALL EXIT
     END



          SUBROUTINE HERMY
    1 (M,N)
     COMMON CA,CB,CC,CD,CE,PI,TPI,TIME,C1,C2,C3,C4,C5,
    1 C6,C7,C8,C9,C10,ROOT(10),WEIT(10       )
    3 , POOT(10),PEIT(10)
9001 FORMAT(9H MMMMMMMM, I3)
     PI=3.14159 26535 89793 23846
     TPI=2.*PI
C
C           GAUSSIAN QUADRATURE
             POOT(1)=0.
C
C           7 POINT QUADRATURE
     IF(N.NE.7) GO TO 7
        POOT(2)=0.4058451513
            POOT(4)=0.7415311855
             POOT(6)=0.9491079123
              PEIT(1)=0.4179591836
              PEIT(2)=0.3818300505
           PEIT(4)=0.2797053914
             PEIT(6)=0.1294849661
   7 CONTINUE
C
C    5 POINT QUADRATURE
     IF(N.NE.5) GO TO 5
         POOT(2)=0.5384693101
             POOT(4)=0.9061798459
              PEIT(1)=0.568888888888888888888888
            PEIT(2)=0.4786286704
             PEIT(4)=0.2369268850
   5 CONTINUE
C
C    3 POINT QUADRATURE
     IF(N.NE.3) GO TO 33
          POOT(2)=0.7745966692
```

```
      PEIT(1)=0.8888888888888888888888
      PEIT(2)=0.5555555555555555
   33 CONTINUE
C
      POOT(3)=-POOT(2)
      POOT(5)= POOT(4)
      POOT(7)=-POOT(6)
      PEIT(3)=PEIT(2)
      PEIT(5)=PEIT(4)
      PEIT(7)=PEIT(6)
C
C     GAUSS HERMITE QUADRATURE
      ROOT(1)=0.
C     7 POINT QUADRATURE
      IF(M.NE.7) GO TO 77
      ROOT(2)=0 8162878828
      ROOT(4)=1.673551628
      ROOT(6)=2.651961356
      WEIT (1)=0.8102646175
      WEIT(2)=0.4256072526
      WEIT(4)=0.05451558281
      WEIT(6)=0.0009717812450
   77 CONTINUE
C
C     5 POINT QUAORATURE
      IF(M.NE.5) GO TO 55
      ROOT(2)=0.9585724646
      ROOT(4)=2.02018287
      WEIT(1)=0.9453087204
      WEIT(2)=0.3936193231
      WEIT(4)=0.01995324205
   55 CONTINUE
C
      IF(M.NE.3)  GO TC 333
      ROOT(2)=1.224744871
      WEIT(1)=1.1816359
      WEIT(2)=0.2954089751
  333 CONTINUE
C
      ROOT(3)=-ROOT(2)
      ROOT(5)=-ROOT(4)
      ROOT(7)=-ROOT(6)
      WEIT(3)=WEIT(2)
      WEIT(5)=WEIT(4)
      WEIT(7)=WEIT(6)
C
C     NORMALIZATION FOR HERMITE POLYNCMIALS
      SQP=1.
```

```
      C1=SQRT(2.*SQP)
      C1=C1/2.
       C2=SQRT(8.*SQP)
      C3=SQRT(48.*SQP)
 C4=SQRT(384.*SQP)
        C5=SQRT(3840.)
         C6=C5*SQRT(12.)
       C7=C6*SQRT(14.)
         C8=C7*SQRT(16.)
     RETURN
      END
 FUNCTION H(I,X)
 COMMON CA,CB,CC,CD,CE
 COMMON PT,TPI,TIME
 COMMON C1,C2,C3,C4,C5
       COMMON C6,C7,C8,C9,C10
 II=I+1
       GO TO (1,2,3,4,5,6,7,8,9)  II
1 H=1.
 RETURN
2 CONTINUE
 H=X/C1
 RETURN
3 CONTINUE
     H2=4.*X*X-2.
 H=H2/C2
 RETURN
4 CONTINUE
  H3=(8.*X*X-12.)*X
 H=H3/C3
 RETURN
5 CONTINUE
       Y=X*X
        H=(16.*Y-48.)*Y+12.
      H=H/C4
 RETURN
6 CONTINUE
       Y=X*X
 H=((32.*Y-160.)*Y+120.)*X
 H=H/C5
 GOTO 10
7 CONTINUE
       Y=X*X
        H=((64.* -480.)*Y+720.)*Y-120.
        H=H/C6
 GOTO 10
8 CONTINUE
9 CONTINUE
```

```
      H=0.
   10 CONTINUE
      RETURN
      END
```

EOF